N 3

```
EEEEEEEEEE XX      XX   AAAAAA   MM      MM PPPPPPPP   LL         EEEEEEEEEE   SSSSSSSS
EEEEEEEEEE XX      XX   AAAAAA   MM      MM PPPPPPPP   LL         EEEEEEEEEE   SSSSSSSS
EEEEEEEEEE XX      XX   AAAAAA   MM      MM PPPPPPPP   LL         EEEEEEEEEE   SSSSSSSS
EE         XX      XX AA      AA MMMM  MMMM PP      PP LL         EE         SS
EE         XX      XX AA      AA MMMM  MMMM PP      PP LL         EE         SS
EE         XX      XX AA      AA MMMM  MMMM PP      PP LL         EE         SS
EE           XX  XX   AA      AA MM  MM  MM PP      PP LL         EE         SS
EE           XX  XX   AA      AA MM  MM  MM PP      PP LL         EE         SS
EE           XX  XX   AA      AA MM  MM  MM PP      PP LL         EE         SS
EEEEEEEE       XX     AA      AA MM      MM PPPPPPPP   LL         EEEEEEEE     SSSSSS
EEEEEEEE       XX     AA      AA MM      MM PPPPPPPP   LL         EEEEEEEE     SSSSSS
EEEEEEEE       XX     AA      AA MM      MM PPPPPPPP   LL         EEEEEEEE     SSSSSS
EE           XX  XX   AAAAAAAAAA MM      MM PP         LL         EE                 SS
EE           XX  XX   AAAAAAAAAA MM      MM PP         LL         EE                 SS
EE           XX  XX   AAAAAAAAAA MM      MM PP         LL         EE                 SS
EE         XX      XX AA      AA MM      MM PP         LL         EE                 SS
EE         XX      XX AA      AA MM      MM PP         LL         EE                 SS
EE         XX      XX AA      AA MM      MM PP         LL         EE                 SS
EEEEEEEEEE XX      XX AA      AA MM      MM PP         LLLLLLLLLL EEEEEEEEEE SSSSSSSS
EEEEEEEEEE XX      XX AA      AA MM      MM PP         LLLLLLLLLL EEEEEEEEEE SSSSSSSS
EEEEEEEEEE XX      XX AA      AA MM      MM PP         LLLLLLLLLL EEEEEEEEEE SSSSSSSS
```

LP

```
**FILE**ID**LPATEST                                C 14




LL          PPPPPPPP      AAAAAA    TTTTTTTTTT  EEEEEEEEEE    SSSSSSSS  TTTTTTTTTT
LL          PPPPPPPP      AAAAAA    TTTTTTTTTT  EEEEEEEEEE    SSSSSSSS  TTTTTTTTTT
LL          PP      PP  AA      AA      TT      EE          SS              TT
LL          PP      PP  AA      AA      TT      EE          SS              TT
LL          PP      PP  AA      AA      TT      EE          SS              TT
LL          PP      PP  AA      AA      TT      EE          SS              TT
LL          PPPPPPPP    AA      AA      TT      EEEEEEEE      SSSSSS        TT
LL          PPPPPPPP    AA      AA      TT      EEEEEEEE      SSSSSS        TT
LL          PP          AAAAAAAAAA      TT      EE                  SS      TT
LL          PP          AAAAAAAAAA      TT      EE                  SS      TT
LL          PP          AA      AA      TT      EE                  SS      TT          ....
LL          PP          AA      AA      TT      EE                  SS      TT          ....
LLLLLLLLLL  PP          AA      AA      TT      EEEEEEEEEE  SSSSSSSS        TT          ....
LLLLLLLLLL  PP          AA      AA      TT      EEEEEEEEEE  SSSSSSSS        TT          ....

FFFFFFFFFF    OOOOOO    RRRRRRRR
FFFFFFFFFF    OOOOOO    RRRRRRRR
FF          OO      OO  RR      RR
FF          OO      OO  RR      RR
FF          OO      OO  RR      RR
FF          OO      OO  RR      RR
FFFFFFFF    OO      OO  RRRRRRRR
FFFFFFFF    OO      OO  RRRRRRRR
FF          OO      OO  RR  RR
FF          OO      OO  RR  RR
FF          OO      OO  RR    RR
FF          OO      OO  RR    RR
FF            OOOOOO    RR      RR
FF            OOOOOO    RR      RR
```

```
C
C       Version 'V04-000'
C
C*************************************************************************
C*                                                                       *
C*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                              *
C*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.               *
C*  ALL RIGHTS RESERVED.                                                 *
C*                                                                       *
C*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
C*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE *
C*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER *
C*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
C*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY *
C*  TRANSFERRED.                                                         *
C*                                                                       *
C*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE *
C*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT *
C*  CORPORATION.                                                         *
C*                                                                       *
C*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS *
C*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.              *
C*                                                                       *
C*                                                                       *
C*************************************************************************
c
c   Example program for LPA11-K  Lab Peripheral Controller
c
c       L P A 1 1 - K  T E S T  P R O G R A M
c
c   This program prompts FOR$INPUT for the set of LPA11-K sample parameters
c   and starts an LPA11-K sweep using those parameters.
C
c                11-Aug-1979
C
c
        integer*2 buffer(20000),rcl(100),iosb(4),device,l
        integer*4 ibuf(50),istat,bufnum,rate,preset,dwell,sampls
        integer*4 strtch,chninc,bffrs,mode,delay,bufsiz,share
        integer*4 input,output,number,comput,rclsiz
        dimension fr(7)
        common /ladata/buffer
        equivalence (iosb(1),ibuf(1))
c
c
c Set some intitial default values for sampling paramaters
c
c Array FR is used to index clock crystal rate for KW11-K
        fr(1)=1000000.
        fr(2)=100000.
        fr(3)=10000.
        fr(4)=1000.
        fr(5)=100.
        fr(7)=60.
c
c Define terminal input and output channels
```

```
TES

!
99

10
20
40
50
60
70
```

```
c
        input=5
        output=6
c
c These are default initial values for interactive paramaters
c
        nmode=-1234      ! microcode mode - load new microcode first time
        rate=1           ! clock counter rate - 1MHz
        preset=-200      ! clock counter preset - 200 ticks
        dwell=1          ! dwell - delay time within each sample sequence
        sampls=1         ! number of samples in a sample sequence
        strtch=0         ! start channel number
        chninc=1         ! channel increment - if zero then random channel list
        bufsiz=1000      ! size of each data buffer
        number=2         ! number of data buffers to use
        bffrs=100        ! total number of buffers to fill
        mode=64          ! sample mode
        delay=10         ! delay before first sample
        device=2hAD      ! sample device type - AD
        comput=0         ! compute load for each buffer
        rclsiz=100       ! size of random channel list
c
c
c Prompt and input SHARE flag
c If share flag is non-zero, the micro-code will not be loaded
c This allows additional copies of this program to be run when the
c LPA11-K is in Multi-Request Mode.  I.E., the first copy of this
c program would be run with the SHARE flag set to 0, causing the clock
c rate to be set, the second and later copies of the program would be
c run with the SHARE flag non-zero, using the previous clock rate set.
c
        write(output,2121)
2121    format(' Share Flag?',$)
        read(input,1002,end=500,err=500)n,share
c
c Prompt for and read in sample paramaters interactively
c
c
c       C L O C K   C R Y S T A L   R A T E
c
10      write(output,1000)rate
1000    format(//' clock rate (',i1,'):',$)
        read(input,1002,err=500,end=500)n,k
1002    format(q,i6)
        if (n .gt. 0 .and. k .lt. 0)goto 24
        if (n .gt. 0 .and. k .ge. 0 .and. k .le. 7)rate=k
c
c       C L O C K   C O U N T E R   P R E S E T
c
        write(output,1004)preset
1004    format(' clock preset: (',i6,'):',$)
        read(input,1002,err=500,end=500)n,k
        if(n .gt. 0 .and. k .lt. 0)preset=k
c
        if (rate .eq. 6 .or. rate .eq. 0)goto 12
        freq=fr(rate)/-preset
        write(output,3000)freq
3000    format('                 clock frquency is ',f12.3,' hertz')
c
c       C O M P U T E   L O A D   P E R   B U F F E R
c
12      write(output,1005)comput
1005    format(' compute load (',i6,'):',$)
        read(input,1002,err=500,end=500)n,k
        if(n .gt. 0 .and. k .ge. 0)comput=k
c
c       D W E L L
c
        write(output,1006)dwell
1006    format(' dwell (',i6,'):',$)
        read(input,1002,err=500,end=500)n,k
        if(n .gt. 0) dwell = k
c
c       N U M B E R   O F   S A M P L E S   p e r   S A M P L E   S E Q U E N C E
c
        write(output,1008)sampls
1008    format(' number of samples (',i6,'):',$)
        read(input,1002,err=500,end=500)n,k
        if(n .gt. 0) sampls=k
c
c       S T A R T   C H A N N E L
c
        write(output,1010)strtch
1010    format(' start channel (',i3,'):',$)
        read(input,1002,err=500,end=500)n,k
        if(n .gt. 0 .and. k .ge. 0 .and. k .le. 128)strtch=k
c
c       C H A N N E L   I N C R E M E N T
c
        write(output,1012)chninc
1012    format(' channel increment (',i3,'):',$)
        read(input,1002,err=500,end=500)n,k
        if(n .gt. 0)chninc=k
        if(chninc .ne. 0)goto 20
c
c       R A N D O M   C H A N N E L   L I S T   S I Z E
c
        write(output,1011)rclsiz
1011    format(' rcl length (',i3,'):',$)
        read(input,1002,end=500,err=500)n,k
        if(n .gt. 0 .and. k .gt. 0 .and. k .le. 100)rclsiz=k
        do 18 ij=1,rclsiz
        rcl(ij)=0
        ik=ij
18      continue
        rcl(ik)=rcl(ik)+'8000'x
c
c       N U M B E R   O F   B U F F E R   A R E A S
c
20      write(output,1013)number
1013    format(' number of buffer areas (',i1,'):',$)
        read(input,1002,err=500,end=500)n,k
        if(n .gt. 0 .and. k .ge. 2 .and. k .le. 8)number=k
c
c       S I Z E   O F   E A C H   B U F F E R
c
        write(output,1015)bufsiz
1015    format(' buffer size (',i5,'):',$)
        read(input,1002,err=500,end=500)n,k
        if(n .gt. 0 .and. k .ge. 10 .and. k*number .le. 20000)bufsiz=k
c
c       T O T A L   B U F F E R S   T O   F I L L
c
        write(output,1014)bffrs
1014    format(' total buffers to fill (',i6,'):',$)
        read(input,1002,err=500,end=500)n,k
        if(n .gt. 0)bffrs=k
c
c       D E L A Y   B E F O R E   S A M P L E   S T A R T
c
        write(output,1016)delay
1016    format(' delay (',i6,'):',$)
        read(input,1002,err=500,end=500)n,k
        if(n .gt. 0)delay=k
c
c       S A M P L E   M O D E
c
c Some typical values for the sample mode are:
c
c       0 - Dedicated Mode
c       64 - Multi-request Mode
c       512 - External Trigger
c       8192 - Dual A/D converters - Serial
c       8224 - Dual A/D converters - Parallel
c
        write(output,1018)mode
1018    format(' sample mode (',i6,'):',$)
        read(input,1002,err=500,end=500)n,k
        if(n .gt. 0)mode=k
c
c       D E V I C E   T Y P E
c
        write(output,1020)device
1020    format(' device type (',1a2,'):',$)
        read(input,1022)n,l
1022    format(q,1a2)
        if(n .le. 0)go to 24
        if(l .eq. 2hAD .or. l .eq. 2hDA .or. l .eq. 2hDI .or. l .eq.
        1 2hDO)device=l
c
c Determine microcode mode from sample mode and device type
c Load new microcode if microcode mode has changed
c
24      if(share .ne. 0)goto 16
        imode=1
        if(iand(mode,64) .eq. 0)imode=2
        if(device .eq. 2hDA .and. imode .eq. 2)imode=3
        if(imode .eq. nmode)go to 16
        call lpa$loadmc(imode,0,istat)
        if(.not. istat)go to 510
        nmode=imode
c
c
c Start lpa11 real time clock at specified rate and preset
c
16      call lpa$clocka(rate,preset,istat)
        if(.not. istat)go to 520
C
C
c Initialize ibuf array for sweep
C
        call ibfint(ibuf,istat,buffer,bufsiz,number)
        if(.not. istat)go to 530
c
c
c Release all the buffers
c
        do 40 i1=0,number-1
        call lpa$rlsbuf(ibuf,istat,i1)
        if(.not. istat)go to 540
40      continue
c
c
c Set channel information for sweeps
c
        if(chninc .ne. 0)call lpa$setadc(ibuf,,strtch,sampls,chninc)
        if(chninc .eq. 0)call lpa$setadc(ibuf,,rcl,sampls,0)
c
c
c Start the sweeps - conditional on what device requested
c
        if(device .eq. 2hAD)call lpa$adswp(ibuf,bufsiz,bffrs,
        1 mode,dwell,,delay,,,istat)
c
        if(device .eq. 2hDA)call lpa$daswp(ibuf,bufsiz,bffrs,
        1 mode,dwell,,delay,,,istat)
c
        if(device .eq. 2hDI)call lpa$diswp(ibuf,bufsiz,bffrs,
        1 mode,dwell,,delay,,,istat)
c
        if(device .eq. 2hDO)call lpa$doswp(ibuf,bufsiz,bffrs,
        1 mode,dwell,,delay,,,istat)
c
        if(.not. istat)go to 550
c
c
c Wait for a buffer to be processed
c
50      bufnum = lpa$iwtbuf(ibuf)
        if(bufnum .lt. 0)go to 100
c
c       *** process data here ***
c
c Go compute bound for some time determined by COMPUT paramater
```

```
c
        do 60 ij=1,comput
        a=sin(ik/5000.)
60      continue
c
c
c Release buffer to be used again
c
        call lpa$rlsbuf(ibuf,istat,bufnum)
        if(.not. istat)go to 540
        go to 50
c
c
c Check for successful completion or error
c
100     if(.not. iosb(1))go to 560
        go to 10
c
c
c
c Various error returns
c
500     call exit
c
510     write(output,2000)istat
2000    format(' error loading microcode  ',i6)
999     nmode=-1234
        goto 10
c
c
520     write(output,2010)istat
2010    format(' error starting real time clock  ',i6)
        goto 999
c
530     write(output,2020)istat
2020    format(' error during "setibf" call  ',i6)
        goto 999
c
540     write(output,2030)istat
2030    format(' error from "rlsbuf"  ',i6)
        goto 999
c
550     write(output,2040)device,istat
2040    format(' error starting ',1a2,' sweep  ',i6)
        goto 999
c
560     itemp=iand(iosb(3),'ff00'x)/256
        write(output,2050)iosb(1),itemp
2050    format(' LPA error - VMS status ',i6,'(D), LPA status ',o3,'(O)')
        goto 999
c
        end
c
c
c Subroutine IBFINT(IBUF,ISTAT,BUFFER,BUFSIZ,NUMBER)
c
```

```
c        IBUF - impure data array for sweeps
c        ISTAT - return status
c        BUFFER - data buffer array
c        BUFSIZ - size of each data buffer
c        NUMBER - number of buffer areas to initialize
c
c IBFINT takes a buffer area, a buffer size and divides it into
c the specified number of individual data buffers.
c
         subroutine ibfint(ibuf,istat,buffer,bufsiz,number)
         integer*4 bufsiz,number
         integer*2 buffer(bufsiz,0:number-1)
         go to (4,4,6,8,10,14,16,18)number
c
4        call lpa$setibf(ibuf,istat,,buffer(1,0),buffer(1,1))
         return
c
6        call lpa$setibf(ibuf,istat,,buffer(1,0),buffer(1,1),
         1 buffer (1,2))
         return
c
8        call lpa$setibf(ibuf,istat,,buffer(1,0),buffer(1,1),
         1 buffer(1,2),buffer(1,3))
         return
c
10       call lpa$setibf(ibuf,istat,,buffer(1,0),buffer(1,1),
         1 buffer(1,2),buffer(1,3),buffer(1,4))
         return
c
14       call lpa$setibf(ibuf,istat,,buffer(1,0),buffer(1,1),
         1 buffer(1,2),buffer(1,3),buffer(1,4),buffer(1,5))
         return
c
16       call lpa$setibf(ibuf,istat,,buffer(1,0),buffer(1,1),
         1 buffer(1,2),buffer(1,3),buffer(1,4),buffer(1,5),
         2 buffer(1,6))
         return
c
18       call lpa$setibf(ibuf,istat,,buffer(1,0),buffer(1,1),
         1 buffer(1,2),buffer(1,3),buffer(1,4),buffer(1,5),
         2 buffer(1,6),buffer(1,7))
         return
         end
```

```
CO

!F
!
!
!
!
$
LO
CO
$
```

0158 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

XALINK
MAR

LABIOLINK
COM

DRMASTER
FOR

LPATEST
FOR

LABIOPEAK
FOR

LABIOSTRT
COM

XAMESSAGE
MAR

XATEST
FOR

LABIOCOM
FOR

LBRDEMO
COM

LABMBXDEF
FOR

LABIOSAMP
FOR

MAILCOMPRESS
COM

LABCHNDEF
FOR

CONNECT
COM

LABIOCON
FOR

LBRDEMO
FOR

PEAK
FOR

DRCOPYBLD
COM

XIDRIVER
MAR

LABIOSEC
FOR

DRSLAVE
FOR

LABIOACQ
FOR

LABIOCOMP
COM

LABIOSTAT
FOR

TESTLABIO
FOR